Here’s the compact one-page summary version of your *System for Building Systems (SBS)* — optimized for executive or client-facing briefs while retaining all key elements from the full architecture.

**🧠 System for Building Systems (SBS)**

Purpose:  
A meta-framework that defines, builds, automates, and evolves other systems with minimal friction, using n8n + PostgreSQL as its automation core.

**🌍 Concept**

SBS is a *self-replicating automation engine* that turns ideas into operations.  
You describe *what a system should do* — SBS handles *how it gets built, automated, and tracked.*

**⚙️ Lifecycle Process**

| **Step** | **Purpose** | **Example Automation** |
| --- | --- | --- |
| 🔍 Define | Capture system purpose, inputs, outputs | n8n webhook form → PostgreSQL |
| 🧩 Design | Map tools, data flow, integrations | Auto-generate markdown “System Design Canvas” |
| 🔧 Build | Construct the working system | Create DBs, folders, APIs via n8n |
| 🤖 Automate | Add triggers + schedules | Cron + event-based flows |
| 🔁 Review | Measure success, log feedback | Telegram/email check-ins + PG logs |

**🏗️ Technical Architecture**

Core Stack

| **Component** | **Tool** | **Role** |
| --- | --- | --- |
| Data Store | PostgreSQL | Single source of truth |
| Automation Engine | n8n | Orchestrates steps + triggers |
| Bot Layer | n8n + Telegram | Sends prompts & updates |
| Scheduler | n8n Cron + PG | Weekly & monthly routines |
| Dashboard |  | Visual system manager |
| Templates | PostgreSQL | Predefined system blueprints |

**🧮 Key Tables**

* systems: primary system registry (name, purpose, category, metadata)
* system\_steps: 5-step lifecycle states
* routines: recurring tasks tied to systems
* system\_templates: reusable blueprints
* system\_logs: audit history and activity log

**🧩 Core Automations**

1. System Spawner:  
Creates lifecycle steps, default routines, and folders when you define a new system.

2. System Orchestrator:  
Tracks step progress and launches correct n8n subflows (Define → Design → Build → Automate → Review).

3. Routine Engine:  
Cron-based flow for daily/weekly reviews and reminders.

4. Telegram Bot:  
Conversational interface — mark steps done, advance flows, get summaries.

**🔔 Real-Time Event Framework**

PostgreSQL emits pg\_notify triggers (system\_update) so n8n instantly reacts to new or updated systems — eliminating polling.

**💬 Example Flow**

1. Add a system: *“Net Worth Tracker.”*
2. n8n auto-inserts lifecycle steps + routines.
3. Markdown canvas generated + folder created.
4. Review reminder scheduled.
5. Telegram bot: *“System ‘Net Worth Tracker’ created. Ready to define inputs?”*

**📊 Output: Self-Improving Systems**

Each SBS-managed system:

* Is version-controlled and auditable.
* Automatically evolves through review cycles.
* Can spawn new systems using stored templates.